International Journal of Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2023.9.3.6

Volume 9, Issue 3
March - 2023

Malware Detection System Using Mathematics of Random Forest Classifier

Akinwole Agnes Kikelomo[1], Yekini Nureni Asafe[2], Ogundele Israel Oludayo[3]

Lecturer[1-3]

[1,3]Department of Computer Technology, [2]Department of Computer Engineering

Yaba College of Technology, Yaba, Lagos

Nigeria

ABSTRACT

Most cyberattacks including data breaches, identity theft, fraud, and other issues, are known to be caused by malware. Some of the malware attacks are categories as adware, spyware, virus, worm, trojan, rootkit, backdoor, ransomware and command and control (C&C) bot, based on its purpose and behaviour. Malware detectors still utilise signature-based approaches to detect malicious software, which can only detect known malware. Attacks by malware pose a serious threat to people's and organizations' cybersecurity globally. These attacks are occurring more frequently and more frequently lately. Over eight billion malware attacks occurred in 2020, up 4% over the previous year, according to a Symantec report. It is crucial that computer users safeguard their computers with a malware detector like an antivirus, anti-spyware, etc. When creating a machine learning model to differentiate between malicious and benign files, it might be challenging to use domain-level expertise to extract the necessary attributes. This research aims to create a malware detector that uses a trained random forest classifier model to find malware and stop zero-day assaults. A dataset (including both harmful and benign software PE header information) was obtained from virusshare.com and used to train the random forest classifier in order to create this malware detector. The Random Forest Classifier generate greater accuracy when compared with other machine learning classifiers, such as KNN (K-Nearest Neighbors), Decision Tree, Logistic Regression etc., the random forest classifier gives a better accuracy of 99.4%. The Classifier model used here will be a better option to use in order to efficiently and effectively detect malware, it shows that the methodology can be utilized as the basis for an operational system for detecting an unknown malicious executable.

Key Words: Cybercriminals, Malware Detection, Malicious Software, Mathematics, Random Forest Classifier.

---

1. INTRODUCTION

The random forest classifier is a branch of mathematics that uses a flexible classification tool to make aggregate predictions using a collection of decision trees trained using the bootstrap method with additional randomness while growing trees by looking for the best features among a randomly chosen feature subset [1].

Data security is the process of preventing unauthorized access, corruption, or theft of digital information. Data security breaches can lead to legal action, hefty fines, and reputational harm to a business or an organization. Protecting data from security risks is more crucial than ever in this modern era [2].

Malware is an intrusive software developed by cybercriminals to steal data and damage or destroy computers and computer systems [3]. Hackers aim malware attacks against individuals, companies, and even governments [4]. The study of computer algorithm that can learn automatically via use and through the utilization of data is known as machine learning. Machine utilizes numerous types of data from host, network and cloud based anti-malware components to improve malware detection. Machine learning algorithm discovers and formalizes the principles that underlie the data it sees [5].

According to statistics, 350,000 new potentially undesirable applications and harmful programmes (malware) are developed each day. Because of this, dealing with malware is a difficult challenge that is also increasing more quickly than the solutions that are being developed for it [6]. The simplest forms of malware detection were found in early antivirus products, which monitors computer systems for specific indicators of compromise, such as exact filenames or signatures, but unfortunately, these detection methods can be easily evaded by a newly created malware with a new signature, or by polymorphic or metamorphic viruses' types of malwares that change their own code each time they propagate thereby ensuring that different versions will have different signatures [7].

The simplest forms of malware detection were found in early antivirus products, which monitors computer systems for specific indicators of compromise, such as exact filenames or signatures, but unfortunately, these detection methods can be easily evaded by a newly created malware with a new signature, or by polymorphic or metamorphic viruses' types of malwares that change their own code each time they propagate thereby ensuring that different versions will have different signatures [7].

This aim of this study is to develop a malware detection system that uses random forest classifier to effectively identify malwares (both zero-day and polymorphic types). The study is limited to development a malware detecting system that implements a trained machine learning model to detect malwares using mathematics of random forest classifier.

2. LITERATURE REVIEW

Majority of malwares were developed from previously existing malware to create new variants, common security tools like virus scanners search for signs in the sample. A virus code can be recognized by its signature, which is a distinctive byte sequence. Metamorphic worms' appearance can be altered by using different strategy to detect malware in a controlled environment or system. Another limitation to malware detection is the failure of the signature-based strategy to identify zero-day attacks. These attacks include threats with evolving capabilities like metamorphic and polymorphic malware [8].

Malware detection technology is used to detect and prevent malware, it is a protection against system intrusions, potential information leaks and system breaches. It is classified into signature recognition, behavior recognition, and feature recognition [9]. Malwares can be analysed using **Static, Dynamic, or** Hybrid analysis [10]; [8]. The detection techniques of malicious software can also be categorized into three main types namely: Signature-based, Anomaly-based, and Heuristic-based techniques [9].

Mathematics of random forest classifier uses decision Tree, random forest algorithm, logistic regression, naïve Bayes, Support Vector Machines etc [1]. Figure 2.1 shows how malware detection can be model having the training and detection phase.

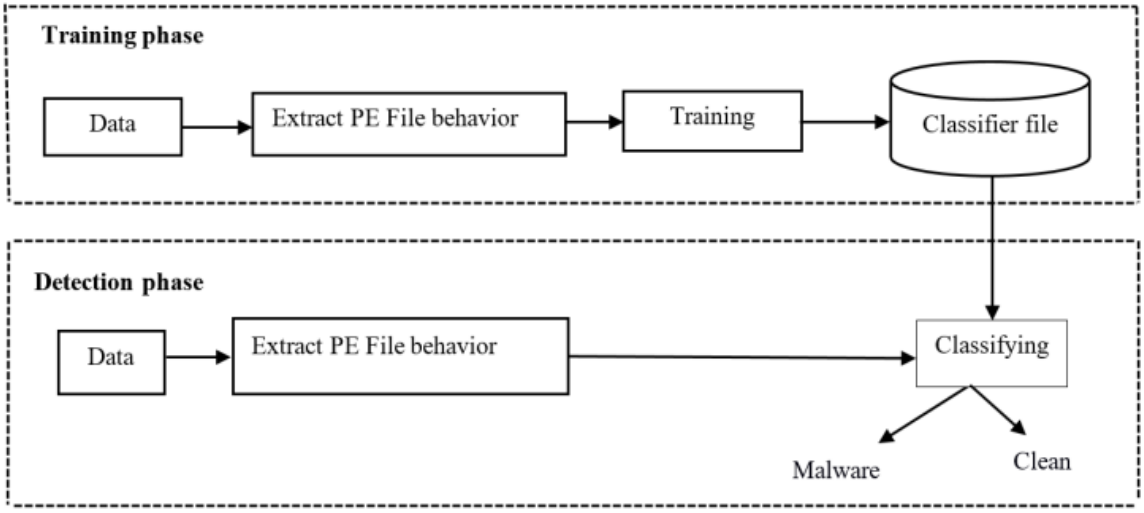

Figure 2.1 Malware Detection Model [11]

Some related works were studied and reviewed during adventure into this study, the reviewed works are described below:

[12] presented a paper on 'Detecting Zero-day Malware' an analysis of the various malware detection methods currently in use and a unique zero-day malware detection model that can effectively distinguish between malicious and benign samples were developed. In the analysis, the authors combined elements of static (signature-based) and dynamic analysis. Without launching or executing the sample, static analysis was utilized to detect any signatures that might have been present and to extract further data, such as the header and packing information. The code was run in a sandboxed environment using dynamic analysis, which provides a thorough analysis report on the sample's behaviour. These two methods offered a more accurate classification model for differentiating between a sample that contains malware and a sample that is not malicious. The authors suggested that in order to fully utilize machine learning algorithms to address the issues and risks facing cybersecurity, they should be modified.

[13] in research titled "Comparing Machine Learning Techniques for Malware Detection" several Machine Learning algorithms were tested and utilized to analyze input PE (Portable Executable) files to establish whether they are malicious or harmless nature. The datasets collected were tested on different models such as Random Forest, Logistic Regression, Naive Bayes, Support Vector Machines, K-nearest neighbors and Neural Networks. Multiple tests were carried out on real data to ascertain the accuracy of the models. The Random Forest Algorithm was the most effective classifier out of all the tested models. The true positive rate was 92%, and the false positive rate was 0.10. Even if the average detection time is not the quickest, this study's examination of the classifiers revealed that Random Forest performs satisfactorily in comparison to other algorithms.

DOI: 10.31695/IJASRE.2023.9.3.6

According [14] presented a paper titled “Integrated static and dynamic analysis for malware detection” For the purpose of detecting malware, this integrated technique incorporated static analysis and dynamic analysis features. The authors' theory was supported by evidence that combining a combination of static and dynamic features will improve detection accuracy over either static analysis alone or dynamic analysis alone. Their research's findings demonstrated that the support vector machine learning method is the most effective at classifying data. Dynamic analysis is superior to code-based static approaches, as shown by the classification findings. Compared to static approaches, the dynamic method is more accurate. It is evident that the integrated strategy improves detection accuracy, in line with the study's goal. The combined technique has a classification accuracy of 98.7%, which is nearly 1.5% better than dynamic analysis.

According to [15] presented a paper titled “Malware Detection Using Machine Learning Algorithms Based on Hardware Performance Counters: Analysis and Simulation”. In this research, a categorization of the HPC-based machine learning methods utilized for malware detection was carried out. The HPC traits that had been used most successfully to detect anomalous activity on various systems were highlighted. The Neural Network (NN) algorithms, including the Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), and Full Order Radial Basis Function (RBF) techniques, were also used to model a number of research from the literature. According to the simulation results, MLP, CNN, and Full Order RBF are accurate to 96.95%, 98.22%, and 98.68%, respectively.

The researched paper [16] titled Detecting Malware based on Analyzing Abnormal behaviors of PE File. This study classified files into normal and malicious code into number of deep learning and machine learning algorithms. The RF and SVM algorithms were selected in accordance with machine learning. Three basic methods—Multi Layers Perceptron (MLP), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM)—were chosen for deep learning techniques. The study proved that the CNN algorithm is much more efficient than the remaining algorithms in all aspects, it had an accuracy of 99.97%.
The research [17] presented a paper on PE-Header-Based Malware Study and Detection. The study presented a simple approach to distinguish between malware and legitimate .exe files. It examined the MS Windows Portable Executable (PE) header properties, and utilizing the structural data that the Microsoft Windows operating system standardizes for the executable, the distinguishing characteristics were extracted from the PE-headers. The methodology used are:

i. Using a Web-Spider, gathering a sizable dataset of malicious and legitimate.exe files from www.downloads.com and www.softpedia.com.
ii. Extracting the characteristics of each header field with a PE-Header-Parser, comparing, and identifying the biggest distinction between malicious and trustworthy.exe files.
iii. Finding the most common icons from the malicious.exe files using an icon-extractor to extract the icons from the PE.

The studies' findings shown that the PE-Header-Based technique distinguishes between benign and malicious executables in less than 20 minutes with a detection rate of more than 99% and less than 0.2% false positives. The outcome demonstrates that malware can be recognized by just examining a few crucial characteristics from PE headers.

According to [18] presented a paper titled "An Efficient Approach for Malware Detection Using PE Header Specifications," In this study, malware programs were discovered, characteristics were retrieved based on the PE file format, and different machine learning algorithms were trained. Based on an analysis of the PE file's structural specification, these features were extracted. The PE header was used to extract the majority of these features. Unlike raw header fields, whose distribution in each data collection can vary, the retrieved traits were those that are typically distinct in malicious and benign applications. By extracting just nine attributes, the new method was able to categorize malicious and benign programs with an accuracy of 95.59%, demonstrating the great competence of the header in the malware detection task.

The authors [19] research on the topic titled “Evaluation of Supervised Machine Learning Techniques for Dynamic Malware Detection”. Data creation, data extraction, classification, and performance metric computation phases made up the dynamic malware detection process utilizing machine learning technique. The data generation phase ran both the benign and malicious PEs in the Cuckoo sandbox’s-controlled environment and generated a JSON file as an execution report. The data extraction phase identified each sample as either benign or malicious and extracted the features from JSON files that described the dynamic behavior of samples. Real malware datasets were produced and used as a result. The findings showed that the TPR and FPR for the instance-based IBk and tree-based J48 ML approaches were both very near to 100% and 0%, respectively. Therefore, these strategies can be viewed as candidates for dynamically constructing an efficient malware detection system for both new and well-known malware samples.

According to [20] presented on The Use of Machine Learning Techniques to Advance the Detection and Classification of Unknown Malware. By employing the RF classifier for feature selection and using cross-validation for data splitting rather than

DOI: 10.31695/IJASRE.2023.9.3.6

the narrative method for data splitting, this study attempted to analyze and expand the work accomplished in a journal that was cited. This resulted in practical performance gains. The cited journal made the data that was used available. 1156 files were gathered, including 984 dangerous files and 172 benign files. These files come in a variety of forms, including.exe, .pdf, and .docx. Accuracy, Precision, Recall, and F1-score were used as performance metrics to evaluate the classifiers. The confusion matrix, which represented the number of real and predicted cases collected by the classifier, was used to determine this. The outcomes showed accuracy gains in comparison to all binary and multi-classifiers. For binary classification, Decision Tree achieved the highest accuracy of 98.2%, while Random Forest achieved the highest accuracy of 95.8%.

3. METHODOLOGY AND DESIGN

The research method and deign started with Collection of Dataset, Preprocessing of Data, Training the Machine Learning Model, and Detection of malicious software respectively. Training the model was developed using the block diagram figure 2 and the dataset features used is showed in Table 1.

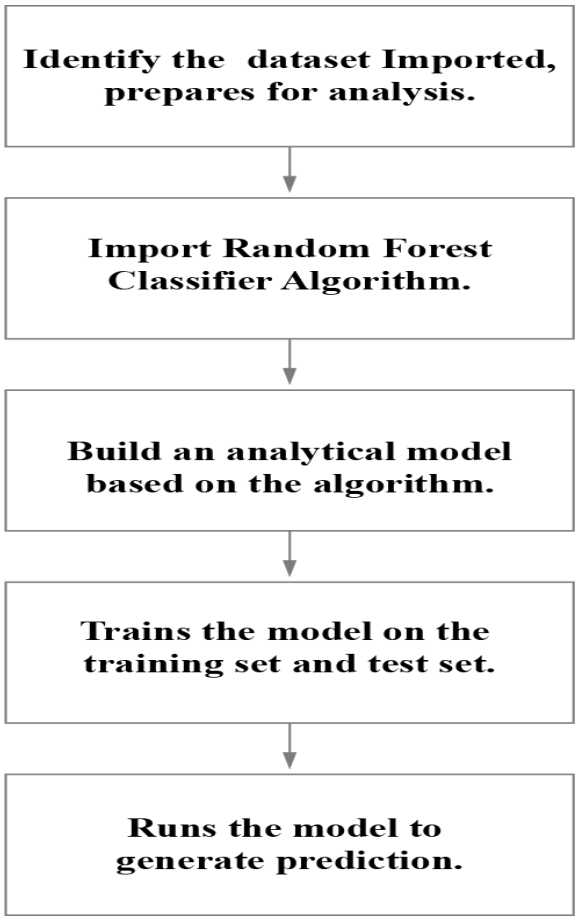

Figure 1: Block diagram showing how the model was trained

Table 1. Description of the features to be used for training

FEATURES	DATA TYPE	VALUE	EXAMPLE
Machine	Integer	0 - 9	322
ImageBase	Integer	0 - 9	4194304
SubSystem	Integer	0 - 9	2
Characteristics	Integer	0 - 9	258
SectionsMaxEntropy	Float	0.0 – 9.0	6.569225321
DLLCharacteristics	Integer	0 - 9	1024
MajorSubsystemVersion	Integer	0 - 9	5
SizeOfOptionHeader	Integer	0 - 9	224

DOI: 10.31695/IJASRE.2023.9.3.6

ResourceMinEntropy	Float	0.0 – 9.0	7.964321949
ResourceMaxEntropy	Integer	0 - 9	3.537939364
SectionMinEntropy	Integer	0 - 9	3.470967855
VersionInformationSize	Integer	0 - 9	16
MajorOpertingSystemVersion	Integer	0 - 9	5
Legitimate	Boolean	0, 1	1

Figure 3, showed the illustration of how features will be extracted for the models training.

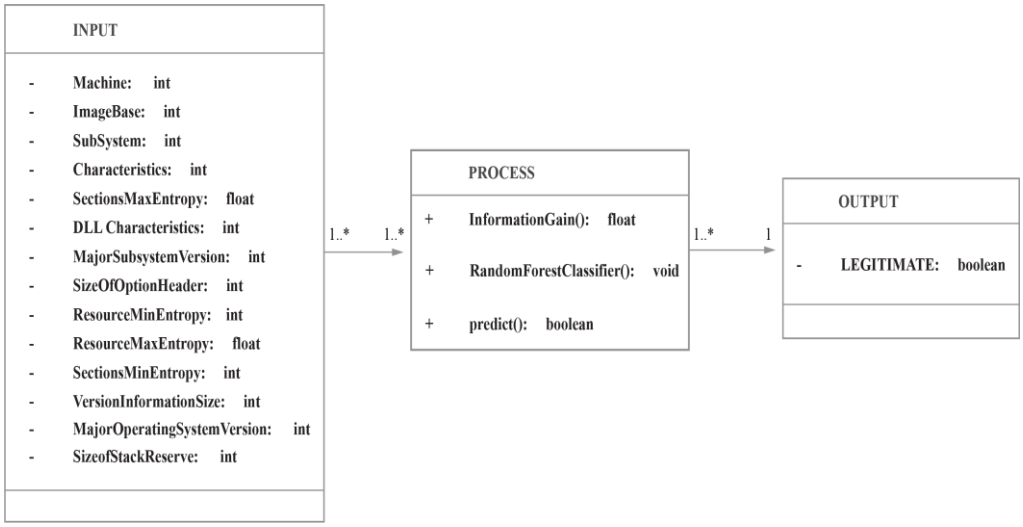

Figure 2: Illustration of How Features will be extracted

The proposed malware detection system will have features that supports its functionality as follow: Scan – scanning allows the malware detection system the search for executable files on a device.

i. View result – this involves viewing scan result.
ii. Delete malwares or threats
iii. Activate/deactivate protection
iv. View history

Flowchart shown in figure 4 depicts the flowchart for program design of the propose malware detection system.

DOI: 10.31695/IJASRE.2023.9.3.6

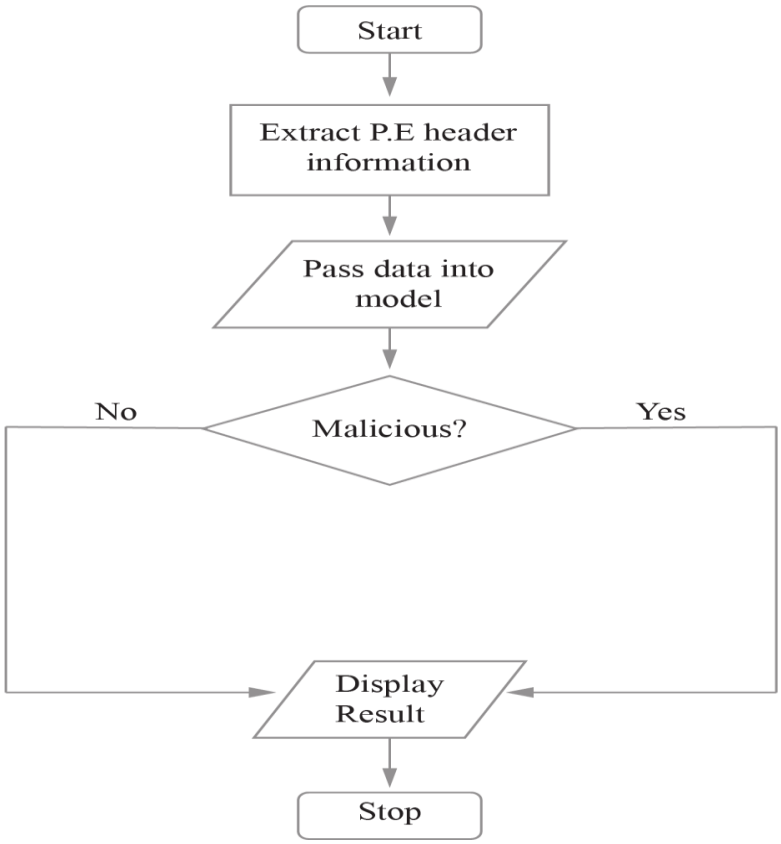

Figure 3: Flowchart Diagram of the Proposed Malware Detection System

4. IMPLEMENTATION

The implementation of the system design involved writing that actual program that represents the proposed system. The proposed system was programmed to detect malwares with the help of the machine learn model and a set of procedures or functions that have been written to extract essential features that are relevant for identifying whether then an executable file is malicious or not. Figure 5-9 are the screenshots of the new system, which is the mobile view of the new system. The Malware detector that was built is an implementation of the malware detection system.

i. **Splash screen:** this is first display of the of the malware detection system, this display allowed the components of the software to load into the devices memory. Figure 4.1 is a screenshot of the malware detectors splash screen.

Figure 4: Splash Screen of the malware detector

DOI: 10.31695/IJASRE.2023.9.3.6

ii. **The malware detector:** This is view of the malware detection software in its idle state.

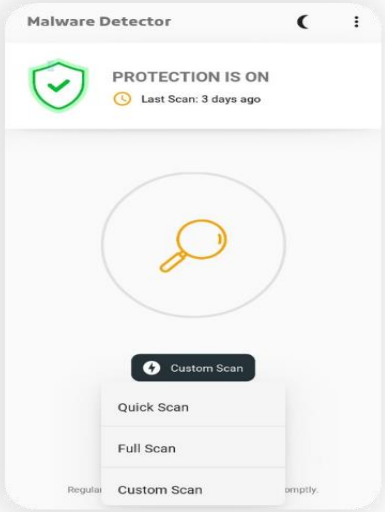

Figure 5: The malware detector of the proposed system

iii. **Scanning:** The basic task of the malware detector is to search its devices internal and external drives (in the case of a full scan), directories or folders to find executable files (i.e., files with the extension of .exe). In the case of a quick scan the malware detector scans common user directories or folders (e.g., Downloads, Documents, Music, and Pictures). Custom scan allows the user to select a particular directory or a particular file for scanning. Figure 4.3 is a screenshot of the malware detector scanning a device.

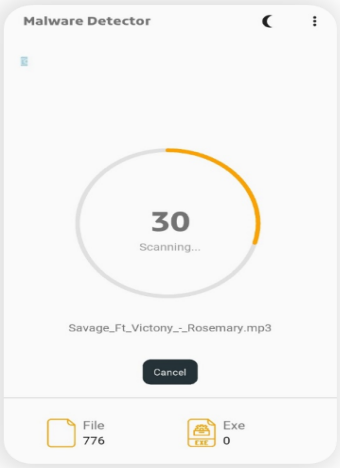

Figure 6. Scanning the device for malicious executable

iv. **Result:** When the malware detector is done scanning a device it examines the executable files found by extracting the PE header information of each executable file found and it passes the details to the trained Random Forest Classifier's predictive model, to identify whether some of the executable files are malicious or not. Figure 8 is a screenshot for the detection of malware, showing the threat that was found. Figure 4.4.2 is a screenshot, showing no threat was found and the device showing the safe state.

DOI: 10.31695/IJASRE.2023.9.3.6

Figure 7: Threat found in malware detector

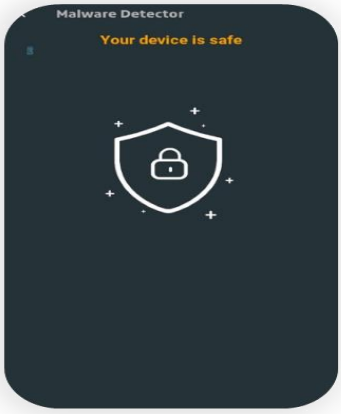

Figure 8: Device at a safe state

5. CONCLUSION

The use of a malware detector has become essential because hackers use malwares to invade people's privacy and gain unauthorized access. Various techniques have been created to evade malware detection and carry out malicious activities unaware. The choice of random forest classifier in detecting malwares was to detect malwares based on the common properties shared. A reliable model was built in this work with the help of a valid dataset which contains PE header information of both malicious and non-malicious files. The malware detection model was trained using the random forest classifier which had an accuracy of 99.4% and a false positive rate of 1.06%. The result of the random forest classifier was compared with the result of some other machine learning classifier like KNN, Logistics regression, gradient booster and so on. The comparison proved the efficiency of the random forest classifier to be the most appropriate for malware detection. The model produce by the random forest classifier was deployed to the malware detector software that was developed. The model acts as the backbone of the malware detection system, which allows it to be able to detect malwares based on their common properties (i.e. detect zero malwares and polymorphic malwares).

REFERENCES

[1] Ronaghan, S. (2018). The mathematics of decision trees, random forest and feature importance in scikit-learn and spark. Towards Data Science, 11.

[2] Kandukuri, S., & Srikanth, G. (2020). A Research Paper on Social Engineering and Growing Challenges in Cyber Security. Think India Journal, 22(41), 11-17.

[3] Ferreira, A. P., Gupta, C., Inácio, P. R., & Freire, M. M. (2021). Behaviour-based Malware Detection in Mobile AndroidPlatforms Using Machine Learning Algorithms. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., 12(4), 62-88.

DOI: 10.31695/IJASRE.2023.9.3.6

[4] Panchal, R. (2021). A review on protection against fileless malware attacks using gateway. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), 7302-7307.

[5] Thosar, K., Tiwari, P., Jyothula, R., & Ambawade, D. (2021, November). Effective Malware Detection using Gradient Boosting and Convolutional Neural Network. In 2021 IEEE Bombay Section Signature Conference (IBSSC) (pp. 1-4). IEEE.

[6] Case, B. (2021). Malware and the Impact of Daily Operations of Public Safety Entities and Law Enforcement (Doctoral dissertation, Utica College).

[6] Musser, M., & Garriott, A. (2021). Machine learning and cybersecurity. Center for Security and Emerging Technology: Washington, DC, USA.

[7] Kwak, G. H., & Hui, P. (2019). DeepHealth: Review and challenges of artificial intelligence in health informatics. arXiv preprint arXiv:1909.00384.

[8] Manavi, F., & Hamzeh, A. (2020, September). A new method for ransomware detection based on PE header using convolutional neural networks. In 2020 17th International ISC Conference on Information Security and Cryptology (ISCISC) (pp. 82-87). IEEE.

[9] Kumar, K. A., Kumar, K., & Chiluka, N. L. (2022). Deep learning models for multi-class malware classification using Windows exe API calls. International Journal of Critical Computer-Based Systems, 10(3), 185-201.

[10] Bawazeer, O., Helmy, T., & Al-hadhrami, S. (2021, July). Malware detection using machine learning algorithms based on hardware performance counters: analysis and simulation. In Journal of Physics: Conference Series (Vol. 1962, No. 1, p. 012010). IOP Publishing.

[11] Van Duong, L., & Do Xuan, C. (2021). Detecting Malware based on Analyzing Abnormal behaviors of PE File. International Journal of Advanced Computer Science and Applications, 12(3).

[12] Abri, F., Siami-Namini, S., Khanghah, M. A., Soltani, F. M., & Namin, A. S. (2019, December). Can machine/deep learning classifiers detect zero-day malware with high accuracy?. In 2019 IEEE international conference on big data (Big Data) (pp. 3252-3259). IEEE.

[13] Moubarak, J., & Feghali, T. (2020). Comparing Machine Learning Techniques for Malware Detection. ICISSP, 10, 0009373708440851.

[14] Shijo, P. V., & Salim, A. J. P. C. S. (2015). Integrated static and dynamic analysis for malware detection. Procedia Computer Science, 46, 804-811.

[15] Bawazeer, O., Helmy, T., & Al-hadhrami, S. (2021, July). Malware detection using machine learning algorithms based on hardware performance counters: analysis and simulation. In Journal of Physics: Conference Series (Vol. 1962, No. 1, p. 012010). IOP Publishing.

[16] Van Duong, L., & Do Xuan, C. (2021). Detecting Malware based on Analyzing Abnormal behaviors of PE File. International Journal of Advanced Computer Science and Applications, 12(3).

[17] Liao, Y. (2012). Pe-header-based malware study and detection. Retrieved from the University of Georgia: http://www. cs. uga. edu/~ liao/PE_Final_Report. pdf.

[18] Rezaei, T., & Hamze, A. (2020, April). An efficient approach for malware detection using PE header specifications. In 2020 6th International Conference on Web Research (ICWR) (pp. 234-239). IEEE.

[19] Zhao, H., Li, M., Wu, T., & Yang, F. (2018). Evaluation of supervised machine learning techniques for dynamic malware detection. International Journal of Computational Intelligence Systems, 11(1), 1153-1169.

[20] Shhadat, I., Hayajneh, A., & Al-Sharif, Z. A. (2020). The use of machine learning techniques to advance the detection and classification of unknown malware. Procedia Computer Science, 170, 917-922.

DOI: 10.31695/IJASRE.2023.9.3.6